Getting User Information with OpenID Connect

Description
In this exercise you'll learn how to request an OpenID Connect ID token and extract the user's information from it.

Estimated Duration
15 minutes

Instructions

Make sure you've completed the first Getting Started exercise, as you'll need the account and setup steps in that exercise to be complete first.

The goal of this exercise is to get a refresh token and use the refresh token to get a new access token. We will be building on the previous exercise where you used the authorization code flow to get an access token. Rather than repeat all the setup steps here, we'll assume you have already created an application and have gone through the authorization code flow at least once.

To get an ID token, you need to add the openid scope to the authorization request. You can also add the profile and email scopes to get more information about the user. Build the authorization URL including those three scopes.

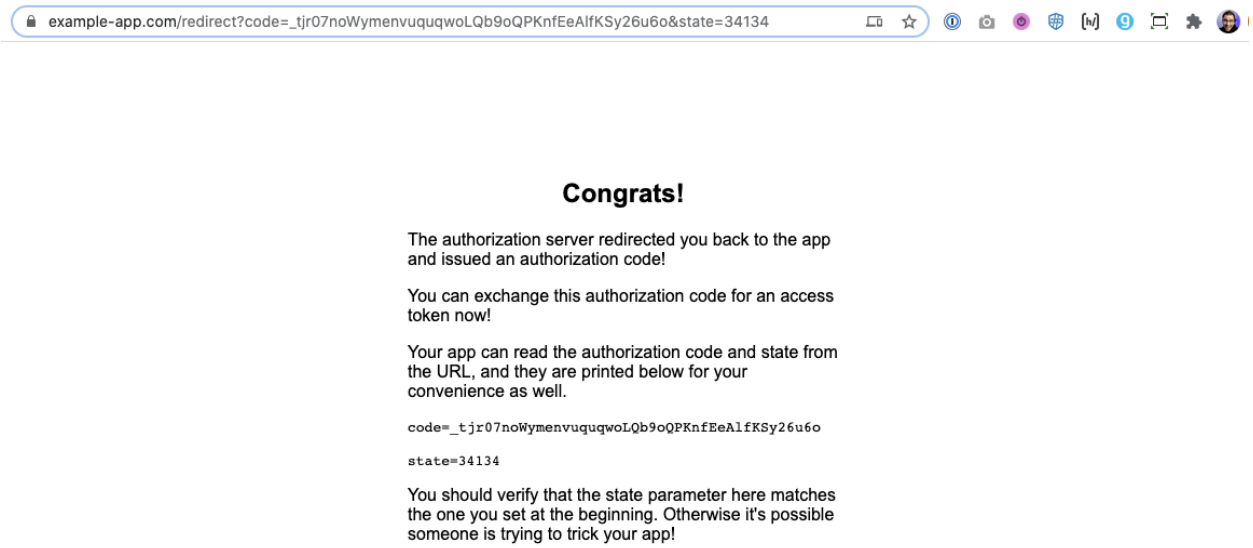
Again you can use the helper tool at https://example-app.com/pkce to generate the Code Challenge and Code Verifier.

> *Note: Copying from PDFs can be error-prone! It is best to re-type everything by hand if you get strange error messages!*

```
https://dev-xxxxxx.okta.com/oauth2/default/v1/authorize?
  response_type=code&
  scope=openid+profile+email&
  client_id={YOUR_CLIENT_ID}&
  state={RANDOM_STRING}&
  redirect_uri=https://example-app.com/redirect&
  code_challenge={YOUR_CODE_CHALLENGE}&
  code_challenge_method=S256
```

Note that we are still using the authorization code flow with PKCE when getting the ID token so that we get it over the back channel, simplifying the process.

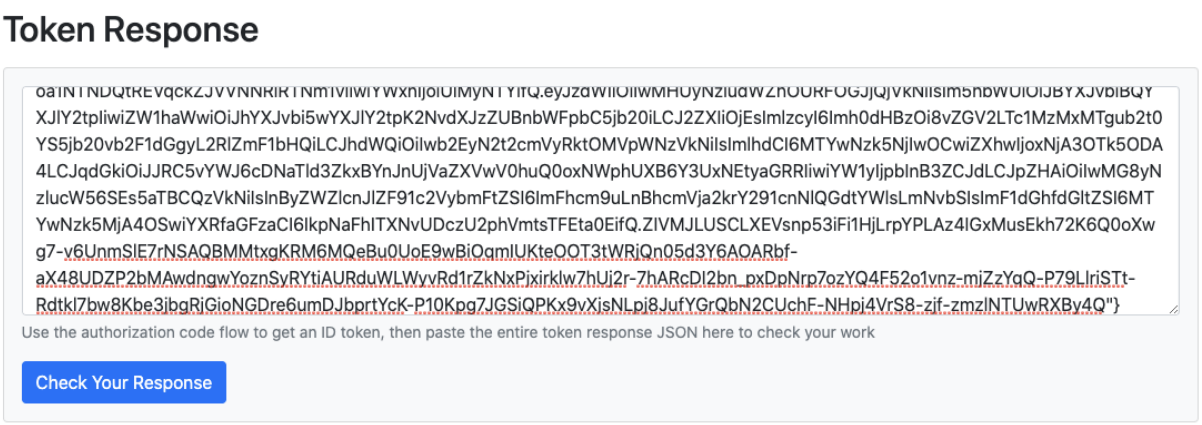
Paste the completed URL into the OpenID Connect exercise (https://oauth.school/exercise/openid/) to check your work. This will double check that you've included the right scope in the request. Once that's confirmed, the "Log In" button will appear. Click that and you'll be taken to the authorization server, and since you're already logged in, you'll be redirected back immediately with an authorization code in the query string.



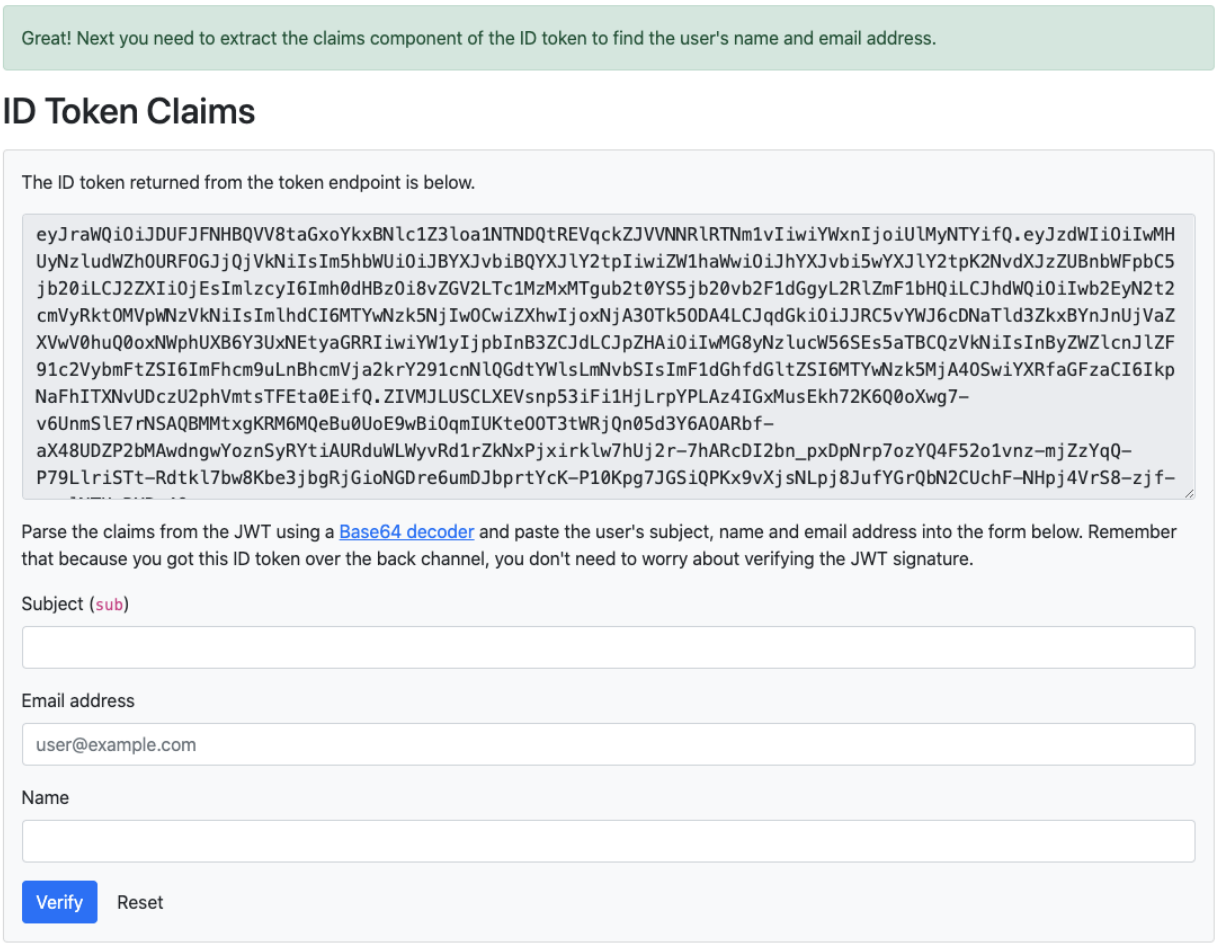
Now you'll need to make a POST request to the token endpoint to get an access token. This request is the same as before. Replace the placeholder values with your own.

```
curl -X POST https://dev-xxxxxx.okta.com/oauth2/default/v1/token \
  -d grant_type=authorization_code \
  -d redirect_uri=https://example-app.com/redirect \
  -d client_id={YOUR_CLIENT_ID} \
  -d client_secret={YOUR_CLIENT_SECRET} \
  -d code_verifier={YOUR_CODE_VERIFIER} \
  -d code={YOUR_AUTHORIZATION_CODE}
```

If everything worked, you'll get back a response that includes an ID token! You may also get an access token if you requested any scopes in addition to the OpenID Connect scopes. Paste the entire token response (not just the access token) into the oauth.school website to check your work.

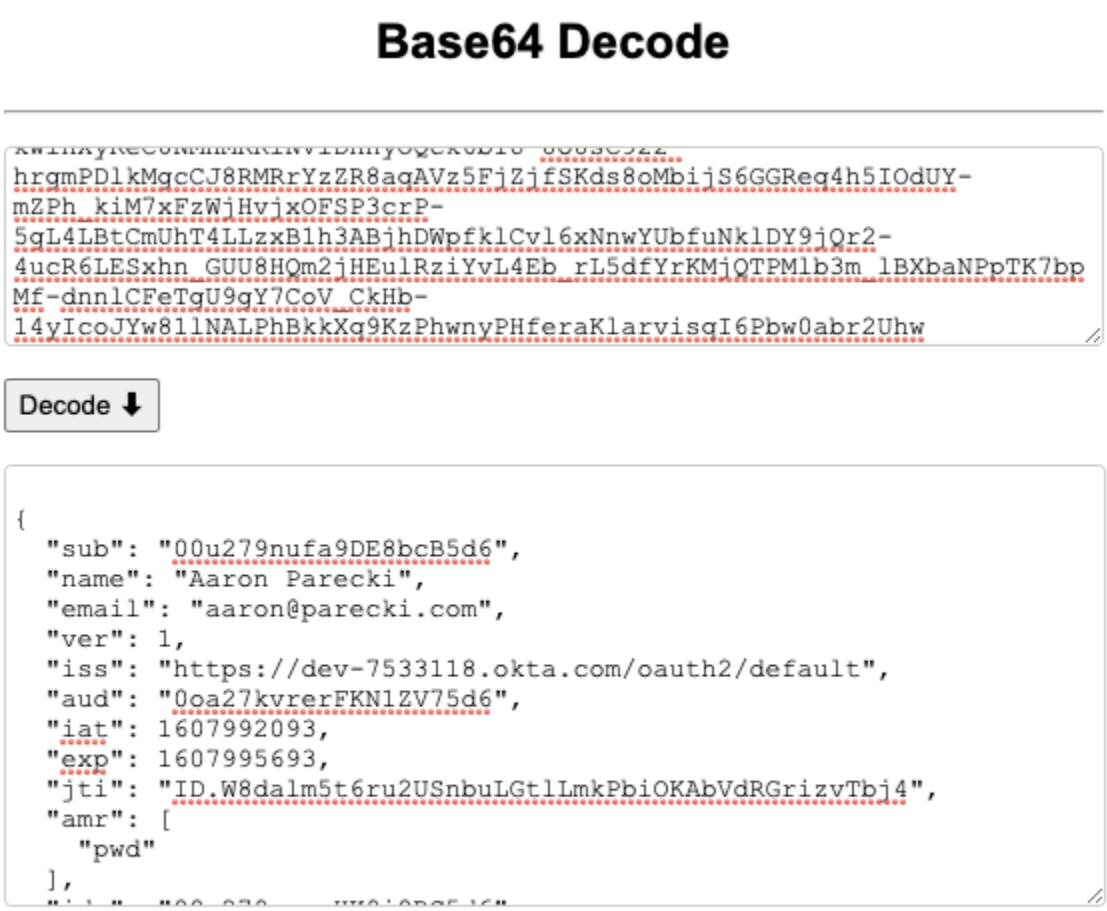


If that worked, you'll be shown the complete ID token and your next job is to parse out the data from it that you care about.



Pull out the claims from the ID token and Base64 decode the data. You can use this website to run the Base64 decode, or you can write code to do that yourself.

https://example-app.com/base64



Copy the sub, name and email and paste them into the testing tool to check your work!